In the Claims

Please amend the following claims:

1. (Amended) A method for temporarily halting execution of a [program's instructions] thread of control while the [program] thread of control is waiting for an event to occur, comprising:

arming an event monitor by identifying at least one event to be monitored;

requesting that the [program] thread of control be halted until any such identified event is observed by the event monitor; and

if execution of the [program] thread of control has been halted,

monitoring, by the event monitor, for an identified event; and

resuming execution of the [program] thread of control subsequent to observation of an identified event by the event monitor.

2. (Amended) The method of Claim 1, further comprising:

halting execution of the [program] thread of control after requesting that the [program] thread of control be halted if an identified event has not yet occurred since the arming.

7. (Amended) The method of Claim [6] 5 wherein a change of access state is observed by monitoring an inter-CPU messaging bus.

11. (Amended) The method of Claim [3] 1, further comprising:

executing an arm instruction to arm the event monitor.

12. (Amended) The method of Claim [11] ~~13~~, wherein the arm instruction identifies [the] an event to be monitored by

identifying at least one memory location; and

recording a physical address of an identified memory location in a working register associated with the thread of control.

13. (Amended) The method of Claim [12] 11, wherein execution of the arm instruction further comprises:

[recording a physical address of the memory location in a working register associated with the program, and]

setting an indicator to a first state which enables the event monitor to monitor for the event, wherein the indicator is set to a second state if the event occurs.

14. (Amended) The method of Claim [13] ~~12~~ wherein the indicator is set to a second state if a change to the memory location whose address is recorded in the working register is observed by the event monitor.

15. (Amended) The method of Claim ~~13~~ wherein, in response to a request to [quiesce] halt the thread of control, execution of the [program] thread of control is halted if the indicator is set to the first state.

16. (Amended) The method of Claim ~~13~~ wherein[, in response to a request to quiesce,] execution of the [program] thread of control is not halted in response to a [the] request to halt the [program,] thread of control if the indicator is set to the second state.

17. (Amended) The method of Claim [13] ~~12~~, wherein execution of the arm instruction further comprises:

loading a value from the identified memory location.

18. (Amended) The method of Claim [13] 11, further comprising:

executing a quiesce instruction to request that the [program] thread of control be halted.

19. (Amended) The method of Claim 18 wherein in the arm instruction and queue instruction are assigned machine codes such that a [program] thread of control utilizing the arm and queue instructions is functional if executed on a machine which does not support the arm and queue instructions.

20. (Amended) The method of Claim 18, further comprising:

flushing [program] thread of control instructions subsequent to the quiesce instruction from an instruction pipeline, upon halting execution of the [program] thread of control.

21. (Amended) The method of Claim [3] 1 wherein the [modification] event is caused by another [program] thread of control.

22. (Amended) The method of Claim 1, further comprising:

while the [program] thread of control is halted,

fetching instructions from the [program] thread of control, and

allowing the fetched instructions to propagate ~~into the instruction pipeline~~.

24. (Amended) The method of Claim 1 wherein the [program] thread of control is executing in a multithreaded environment.

27. (Amended) The method of Claim 1 further comprising the step of:

upon halting execution of the [program] thread of control, setting a timer to time a predetermined time interval, and starting the timer; and

resuming execution of the [program] thread of control upon expiration of a timer.

28. (Amended) The method of Claim 27, further comprising:

stopping the timer if execution of the [program] thread of control is resumed due to observation of the event by the event monitor.

29. (Amended) The method of Claim 1, wherein halting execution of the [program] thread of control results in a reduction of power consumption.

30. (Amended) The method of Claim 1, wherein halting execution of the [program] thread of control allows other executing [programs] threads to control to utilize available resources.

31. (Amended) ~~A~~ system for temporarily halting execution of a [program's instructions] thread of control while the [program] thread of control is waiting for an event to occur, comprising:

an event monitor which is armed via identification of an event to be monitored; and

an execution scheduler, responsive to the event monitor, which, upon a request that the [program] thread of control be halted until the event is observed by the event monitor, halts execution of the [program] thread of control if the event has not yet occurred since the event monitor was armed, and which resumes execution of the [program] thread of control upon observation of the event by the event monitor.

(Amended) The system of Claim ~~32~~ 34, further comprising:

a working register associated with the [program] thread of control, into which a physical address of the memory location is stored upon arming of the event monitor; and

an indicator associated with the [program] thread of control, [which is] the indicator being set to a first state upon arming the event server, causing the event monitor to monitor for the event, and [which] the indicator being set to a second state by the event monitor upon a change to the memory location whose address is recorded in the working register.

~~38.~~ (Amended) The system of Claim ~~37~~ 39, wherein a [lock] value is loaded from the identified memory location upon storing the memory location's address in the working register, such that a determination may be made as to whether the memory location's state has changed after arming the event monitor and before halting the [program's] thread's execution.

39. (Amended) The system of Claim [38] 31 wherein an executing arm instruction arms the event monitor.

~~40.~~ (Amended) The system of Claim [39] ~~31~~ 33, wherein an executing quiesce instruction halts execution of the [program only if the flag is set] thread of control until the identified event is observed.

42 34
41. (Amended) The system of Claim [40] 32 wherein a change to the memory location comprises a change of state of the memory location, caused by another [program] thread of control, from shared to exclusive.

43 34
42. (Amended) The system of Claim [40] 32, wherein a change to the memory location comprises a write operation to the memory location, observed by monitoring the address on a memory write bus.

44 41
43. (Amended) The system of Claim 40, further comprising:

an instruction flusher for flushing [program] thread of control instructions subsequent to the quiesce instruction from an instruction pipeline if the [program] thread of control is halted.

45 41
44. (Amended) The system of Claim 40 [43], wherein, while the [program] thread of control is halted, [program] thread of control instructions are fetched and allowed to propagate into the instruction pipeline.

---

48 33
47. (Amended) The system of Claim 31 wherein the [program] thread of control is executing in a multiprocessor environment.

49 33
48. (Amended) The system of Claim 31 further comprising:

a timer associated with the [program] thread of control such that a timer, upon the halting of its associated [program] thread of control, is set to time a predetermined time interval, and started, wherein execution of the [program] thread of control is resumed upon expiration of a timer; and wherein the timer is stopped if execution of the [program] thread of control is resumed due to observation of the event by the event monitor.

~~Please add the following new claims:~~

sub B11
~~49.~~ A system for temporarily halting execution of a thread of control while the thread of control is waiting for an event to occur, comprising:

event monitoring means;

arming means for arming the event monitoring means by identification of at least one event to be monitored;

requesting means for requesting that the thread of control be halted until any such identified event is observed by the event monitoring means; and

halting means for halting the thread of control responsive to the requesting means, wherein if execution of the thread of control is halted, execution of the thread of control is resumed subsequent to observation of an identified event by the event monitoring means.

50. An electronic circuit for temporarily halting execution of a thread of control while the thread of control is waiting for an event to occur, comprising:

an event monitor circuit, for monitoring for at least one event identified by the thread of control;

a quiesce logic circuit, which, responsive to the event monitor circuit and to a request from the thread of control to quiesce, temporarily halts execution of the thread of control, and which, responsive to the event monitor circuit upon observation of at least one identified event, resumes execution of the halted thread of control.

51. The electronic circuit of Claim 50, further comprising:

a flushing circuit which, upon halting execution of the thread of control, flushes the thread's instructions from an instruction pipeline subsequent to the request to quiesce.

52. The electronic circuit of Claim 50, further comprising:

a mapper selection circuit which selects, for mapping, at least one instruction in at least one instruction buffer associated with at least one non-quiescing thread of control.

53. The electronic circuit of Claim 50, further comprising:

a fetch selection circuit which selects, for fetching, at least one program counter associated with at least one non-quiescing thread of control.

54. The electronic circuit of Claim 50, further comprising:

a timer circuit associated with the thread of control, wherein, upon the halting of its associated thread of control, the timer circuit is set to time a predetermined time interval and started, wherein the quiesce logic circuit, responsive to the timer circuit, resumes execution of the thread of control upon expiration of the interval.

30
55. The method of Claim 21 wherein the another thread of controls executing in a different CPU.

31
56. The method of Claim 21 wherein the another thread of controls executing in a different TPU.

32
57. The method of Claim 18 wherein instructions are executed out of order.

Respectfully submitted,

HAMILTON, BROOK, SMITH & REYNOLDS, P.C.

By _____
Gerald M. Bluhm
Registration No.: 44,035
Telephone (781) 861-6240
Facsimile (781) 861-9540

Lexington, Massachusetts 02421-4799
Dated: 5/11/00